AN ASSET OVER A MULTI-TIERED NETWORK, PARTICULARLY TO A TARGETED NODE

12. SYSTEM AND METHOD FOR ADJUSTING THE DISTRIBUTION OF AN ASSET OVER A MULTI-TIERED NETWORK

13. SYSTEM AND METHOD FOR BRIDGING ASSETS TO NETWORK NODES ON MULTI-TIERED NETWORKS

14. METHOD AND SYSTEM FOR DEPLOYING A CHANGED ASSET OVER A MULTI-TIERED NETWORK

15. SYSTEM, METHOD, AND DATA STRUCTURE FOR PACKAGING ASSETS FOR PROCESSING AND DISTRIBUTION ON MULTI-TIERED NETWORKS

ABSTRACT

Abstract 1. EE

An extended environment data structure that is part of a digital asset that is capable of being transmitted over one or more multi-tiered networks. The data structure has one or more common descriptors, that provide a unique identification of the digit asset on the networks; one or more asset dependency descriptors that identify one or more associated digital assets, associated digital assets are associated with the digital asset by means of a joint membership as parts of a whole; and has one or more target server dependencies descriptors that identify a base execution environment on one or more target computers. In an alternative embodiment, one or more EIS server dependencies descriptors are included that identify an EIS execution environment on the respective EIS from which the asset resides. In other alternative embodiments, other descriptors are included in the extended environment data structure.

Abstract 2. Discovery

A system, method, article of manufacture, and a computer program product identifies (discovers) member objects of one or more computer system parts in an Enterprise Information System (EIS) while establishing at least one topographical relationship among the member objects in order to create digital assets. The invention begins by traversing one or more computer file systems of the EIS to find one or more of the member objects. For each member object found, a digital asset identifier of the found member object is placed in an intermediate representation. The intermediate representation is a graph with nodes and edges. Each of the digital asset identifiers corresponds to one of the nodes of the graph. The edges

represent the topographical relationship. A digital asset is created from the member object by placing the member object in a logic/data section of the digital asset and attaching an extended environment data structure to the logic/data section. The digital asset is stored in an asset inventory container object. This may be repeated for each found member object until the intermediate representation fully describes the computer system part and the asset inventory container object is a complete inventory of the digital assets of interest in the computer system part.

Abstract 3. Export

Member objects of one or more computer system parts in an Enterprise Information System (EIS) are identified and categorized for export to either a packaging process or to another computer system over tiers of one or more networks. An intermediate representation of one or more parts of a computer system while applying one or more context rules to determine a context of the parts. The context may be either a standard specified context or a non-specified context. If a standard specified context is determined, a directed search is performed to acquire any of the set of runnable and/or non-runnable member objects in one or more locations in an Enterprise Information System (EIS) as identified in the intermediate representation and specified by the context. If a non-specified context is determined, an implicit traversal search is performed for any of the set of runnable and/or non-runnable member objects in one or more locations in an Enterprise Information System (EIS) identified in the intermediate representation. One or more of the set of runnable and/or non-runnable member objects are then accessed at their respective locations in the EIS. A preliminary package specification is made for the accessed set of the runnable and/or non-runnable member objects. Digital assets in an asset inventory that correspond to the respective runnable and non-runnable member objects, are listed in the preliminary package specification and are updated by adding one or more export descriptors to the extended environment of the respective digital assets.

Abstract 4. CDS

A computer system and method is used as a server connected to one or more networks through one or more network interfaces. The server, called a component distribution server (CDS) system, comprises a package specification process that receives one or more packages from the network. These received packages are subparts of one or more application programs from one or more enterprise information systems (EIS). The packages have one or more assets. Each asset has an asset type and two or more asset layers. A first asset layer is a logic/data layer with information that embodies the function of the asset (asset function). The second asset layer is an extended environment layer that is a subset of the EIS and that has portions of the EIS necessary to support the respective logic/data layer. The CDS system also has a process adapter process that translates one or more of the assets layers so that the asset can perform the asset function on one or more target base environments of one or more target computers. Further, the CDS system has a target process that changes one or more of the layers of the asset in order to provide specific information for one or more specific target computers.

Abstract 5. Package

A novel package structure, a system for using this package structure, and a program product containing this package structure is disclosed. The package structure can be stored in one or more memories (e.g., as an article of manufacture) and/or distributed over one or more networks. The package structure has one or more first assets, each with a first logic/data part and a first extended environment part. These first assets can have a relational data asset type or a static content asset type. The package structure also has one or more second assets, each having a second logic/data part and a second extended environment part. The second assets have a presentation component asset type, typically used to generate content. In alternate preferred embodiments, the package structure also has one or more third assets, each with a third logic/data part and a third extended environment part. The third assets have a transaction component asset type, relational data asset type, or a static content asset type (if the first asset has a relational data asset type). In another alternative preferred embodiment, one or more forth assets exist. The four assets are selected so that the package has at least one asset from each of the following asset types: relational data, presentation component, transaction component, and static content.

Abstract 6. Transactional deployment – J2EE

A computer system, method, article of manufacture, and computer program product for transactional deployment of one or more components over a multi-tier network is disclosed. The system has one or more J2EE application servers stored on one or more memories of the system and capable of being executed by one or more central processing unit (CPUs). One or more J2EE applications can be executed on the J2EE application servers. One or more J2EE application containers are contained within the J2EE application. One or more J2EE application container components are contained within the respective J2EE application containers. One or more delivered J2EE components are delivered to the J2EE application server over one or more tiers of one or more the network. There are one or more logical connections to one or more databases located on the networks. In addition, there is a sphere of control encompassing the J2EE application server, the J2EE applications, the J2EE application containers, the J2EE application container components, the delivered J2EE components, and the logical connections. The sphere of control manages a transactional deployment of the delivered J2EE components and an update of the databases to keep the data consistent with the J2EE application.

Abstract 7. Deploy – Manifest

An exemplary method and/or exemplary embodiment of the present invention distributes an asset to a multi-tiered network node. An asset may represent network and/or application components (e.g., data, objects, applications, program modules, etc.) that may be

distributed among the various resources of the network. In an embodiment, a pending notice is received from a distribution server. If the notice indicates that at least one asset is pending (i.e., awaiting deployment), an asset descriptor manifest is received from the distribution server. The asset descriptor manifest identifies at least one asset to be deployed to the node, and includes an offset associated with the asset identifier. The asset descriptor manifest is stored in a memory on the node. A fragment, associated with the asset, is received and stored in the memory. The offset associated with the asset is marked with the end of the fragment, and another fragment, beginning at the offset, is then received. Additional fragments are received, and the offset updated, until the entire asset is deployed to the node. In an alternative embodiment, the entire asset is received in the first fragment. In a further embodiment, multiple assets are received in the first fragment.

Abstract 8. Deploy – Adapter

A method and system for deploying assets to multi-tiered network nodes. An asset may represent network and/or application components (e.g., data, objects, applications, program modules, etc.) that may be distributed among the various resources of the network. In one embodiment, a target node's environment may be adjusted before an asset is deployed to that target node. In an alternative embodiment, a target deployment adapter, associated with the asset, may be selected and deployed with the asset in order to allow the asset to operate in the target node environment. An implementation class, associated with the asset, may be inserted into the target node environment. An altered target deployment descriptor may also be inserted into the target node environment.

Abstract 9. Translation/Process

The present invention provides a system and method for translating an asset for distribution to a multi-tiered network node. An asset may represent network and/or application components (e.g., data, objects, applications, program modules, etc.) that may be distributed among the various resources of the network. In an embodiment, an asset has a

logic/data section and an extended environment section. The logic/data section defines a function of the digital asset along with the asset's type, while the extended environment section supports the function of the logic/data section within at least one source environment. The asset type is determined and a process asset adapter, associated with the asset type and a target environment, is selected. The asset is then translated into a processed asset having a processed extended environment section supporting the function of the logic/data section in the target environment.

Abstract 10. Synchronization

An exemplary method and/or exemplary embodiment of the present invention synchronizes an asset over a multi-tiered network. An asset may represent network and/or application components (e.g., data, objects, applications, program modules, etc.) that may be distributed among the various resources of the network. Synchronization addresses the restoration of asset coherency in a distributed system, i.e. bringing changes made to assets on one distributed node into harmonization with changes made to assets on another distributed node. In an embodiment, a synchronization call having a data argument and an asset type is received, an adapter associated with the asset type is selected, and the data argument is passed to the adapter. The asset type is determined, as well as a table associated with the asset type. A synchronization information object is retrieved from a target environment on a target node, and a synchronization asset is created based on the synchronization information. A connection is established between the target node and the asset's original source node, and the synchronization asset is sent from the target node to the source node.

Abstract 11. Target

A method of operating a computer system for targeting one or more digital assets on a distribution server connected to one or more networks so that the digital assets are compatible with one or more target nodes connected to the networks includes examining the one or more digital assets to determine an asset type of the digital asset and, if the asset type is Relational Data (RD), retrieving one or more where clauses of the digital asset. The method further

includes executing a token replacement operation on the where clause to create a transformed where clause and running a query on one or more tables specified in the digital asset using the transformed where clause, the query returning one or more returned records and the returned records correlating with the target node. The method further includes storing the returned record in the digital asset.

Abstract 12. Adjust

The present invention provides a system and method for adjusting the distribution of an asset over a multi-tiered network. An asset may represent network and/or application components (e.g., data, objects, applications, program modules, etc.) that may be distributed among the various resources of the network. In an embodiment, a performance metric is received and input to a performance model. The model determines a network optimization and at least one change requirement based on the metric. A package specification is changed to reflect the requirement, which may specify that at least one asset should be packaged in at least one package. The change requirement is implemented and the package is distributed over the network.

Abstract 13. Bridging

An exemplary method and/or exemplary embodiment of the present invention provides a system and method for bridging an asset over a multi-tiered network. An asset may represent network and/or application components (e.g., data, objects, applications, program modules, etc.) that may be distributed among the various resources of the network. Generally, communications can be maintained between executable assets residing on different network nodes by bridging the execution context of the two nodes. In an embodiment, a mapping layer can be generated for assets that have run-time dependencies; the mapping layer uses a distribution system to bridge the execution context of a first environment with that of a second environment. The asset executing in the first environment is able to access another resource

located in the second environment, even though the asset does not have local access to the resource in the second environment. A fault is detected when at least one asset deployed on a local node attempts to access at least one resource on a remote node through an application programming interface. The fault is then handled appropriately.

Abstract 14. Stream

A method for distributing changes to digital assets across a network includes determining an asset type of a first digital asset and comparing the first digital asset to a prior digital asset to determine one or more deltas, the prior digital asset being a prior version of the first digital asset and the delta being a difference between the first digital asset and the prior digital asset. The method further includes evaluating the one or more of the deltas with one or more criteria to determine if the one or more delta assets should be created, the delta asset being a second digital asset containing the respective delta, the criteria determined by the asset type. The method further includes that if the delta meets the criteria, creating the delta asset, and marking the delta asset as a first delta asset of the first digital asset.

Abstract 15. EEP

The present invention provides a system, method, and data structure for packaging assets for processing and distribution over a multi-tiered network. An asset may represent network and/or application components (e.g., data, objects, applications, program modules, etc.) that may be distributed among the various resources of the network. In an embodiment, the package structure includes at least one representation of an asset having a logic/data portion and an asset extended environment portion, and a package extended environment that includes package information associated with at least one asset.